Rules for Component Evaluations under CCEVS

Background

This policy is issued as a supplement to Scheme Policy Letter #2 ("Reuse of Previous Evaluation Results and Evidence", 4 March 2002). The focus of this policy statement is for the case where a developer produces several products or product lines that all contain an identical instantiation of a significant security-critical component (such as a security micro-kernel). This policy establishes a means whereby this component could be evaluated in isolation, and then subsequent evaluations of products that contain this component could theoretically treat it as a black box, and there would be no need of re-analysis or retesting of its internals.

This policy describes the rules under which the evaluation of such a component is to take place, together with the rules for reusing the results of that evaluation as part of a subsequent evaluation. ***It covers only the case where the developer of the component is also the developer of the product(s) containing that component.***[1]

Rules for Conducting the Component Evaluation

In a typical evaluation, the evaluation evidence and ETR is proprietary and is therefore kept within the confines of the evaluation; the only released outputs are the ST and the VR. A *component evaluation* differs from a usual evaluation in that there will also be a new piece of evidence specially created for use in the subsequent evaluation(s). (Because this policy is aimed at cases where the developer of the component is also the developer of the subsequent product, it is presumed that the component evaluation evidence will be available for use in the subsequent evaluation(s).)

A component evaluation approach is defined by the following rules:

1. The ST of the component must clearly define its environment. Note this environment might be part of the product or part of the product's environment, as shown in Figure 1.

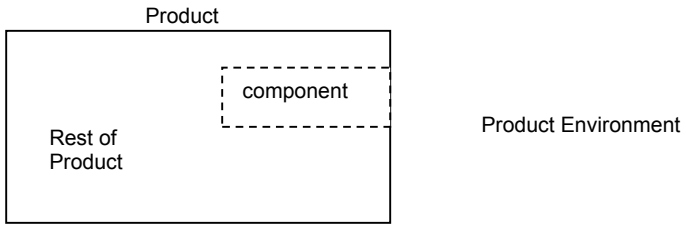

Figure 1

2. In addition to the standard evaluation outputs (Security Target, Validation Report), the component evaluation will also define a *Composition Requirements Definition* (CRD), which specifies all of the interactions the component will have with its environment, along with its assumptions about (or expectations of) its environment. This provides a specification of the dashed line shown in Figure 1.

[1] This policy does not apply to the case of the developer of the product being different from the developer of the component; the Scheme believes prudence requires that the technical issues involved in reusing evaluation results must be firmly established and successfully addressed before increasing the scope of the approach to accommodate the proprietary issues that would be involved in the cases of different developers.

The contents of the CRD specification are:

- The syntax of the security-relevant interfaces of the component.
- The semantics of any security functions the component provides and of any properties that it guarantees.
- The error conditions the component can generate and the semantics of each of these with respect to security properties and guarantees.
- The security assumptions that the component makes about its environment. This includes a list of properties of its environment that it assumes to be invariant, as well as the list of properties it maintains as invariant. It should enumerate all the security properties (functions and policies) upon which it depends and consumes from other components, and what trust assumptions it makes about those components. Also included is a description of the environment's role in the protection and non-bypassability of the component.
- The vulnerabilities that had been looked for in the evaluation of the component, and the environmental assumptions that were made during the vulnerability analysis. This includes both the methodology that was used, as well as the resulting findings.
- A description of the process by which the CRD is made available to developers of products that will incorporate the component.

The description of the security-relevant interfaces is on par with the description called for by the functional specification requirements. As such, this content might be already within the evaluation evidence (if the component boundary is the same as the TSF boundary), in which case the CRD would simply be an index to its content within the other evaluation evidence (identifying which of the interfaces described in the functional specification are security-relevant and referencing them). However, it might be the case that the component's interfaces are not TSF interfaces and would therefore not be included in the functional specification, and so these details would have to be provided. As with all evaluation evidence, there is no need to create it specifically for an evaluation if the necessary content already exists.

3. The CRD will remain on file with the CCEVS.
4. The following ETR sections and corresponding evaluation evidence from the component evaluation will be made available (by the sponsor of the component evaluation) to the parties involved in the evaluation of the product: the component interface specification, the component testing evidence, the component vulnerability analysis.
5. The results of the component evaluation will not be listed on the CCEVS Validated Products List as an evaluated product; it will instead be listed as an evaluated component[2].

Rules for Reusing Component Evaluation Results in Subsequent Evaluations

The subsequent evaluation of a TOE (of which an evaluated component is a part) is conducted much like any other evaluation. However, its reuse of prior evaluation activities and results, as defined in the component's CRD, allows the component to be viewed as a black box ("Component A" in Figure 2); the TOE evaluation will focus on the remaining components of the TOE.

[2] This policy is restricted to evaluated components; it does not cover cases where evaluated *products* are to be reused in subsequent evaluations (e.g. an evaluated OS later used in conjunction with a trusted application). The use of evaluated products is considered a different *composition* issue, which will be addressed separately.

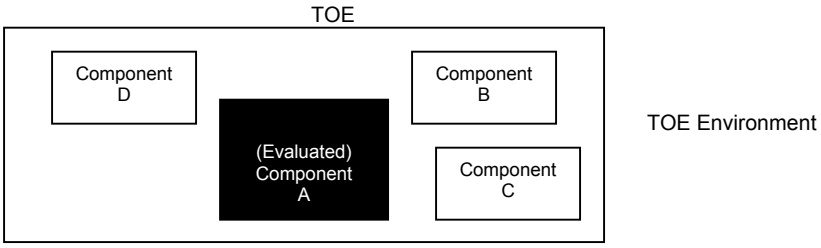

Figure 2

This reuse of prior evaluation results is constrained by the following rules:

1. The evaluated component is identified as a subsystem or module in the TOE's High-Level Design or Low-Level Design.
2. The evaluated component is configured and used in its evaluated configuration (as defined by the component evaluation).
3. The ETR will note that the evaluation included the results of a subsystem evaluation, citing the CRD that was used to replace the analysis and testing of the subsystem.
4. The ETR will use the CRD from the component evaluation to assess the following:
    - The semantics of any security functions or properties provided by the environment or by any previously evaluated components upon which the TOE relies. In order to evaluate the composition of the TSFI with the environment, it is necessary to understand what the TSFI thinks the environment is doing when it calls out.
    - A description of how it deals with error conditions and/or undefined output from any relevant previously evaluated components.
    - Verification that the TOE meets all of the security assumptions about (and expectations of) the environment that are specified in the component's CRD. This includes a list of properties of its environment that it assumes to be invariant, as well as the list of properties it maintains as invariant. It should enumerate all the security properties (functions and policies) upon which it depends and consumes from other components, and what trust assumptions it makes about those components. This also includes a description of the how the TOE fulfills the needs of the component to enforce any protection and non-bypassability of the component.
    - The set of test cases that an evaluator of the TOE could use to determine whether the component supplies the security functions and properties required by the TOE. This could be achieved by testing (in accordance with current assurance requirements) the interfaces invoked by the component. The test documentation should include a list of tests whose successful execution will demonstrate that the environment (or any relevant evaluated component) fulfils its security obligations.
    - Using the evaluated component's vulnerability analysis as input to the TOE's vulnerability analysis, an enumeration of the types of vulnerabilities in the evaluated component that the evaluator believes could cause problems in the TOE, if they existed.

5. The ETR sections and corresponding evaluation evidence from the component evaluation (component interface specification, subsystem testing evidence, subsystem vulnerability

analysis) will be updated as necessary to reflect any relevant interpretations that have been issued.

6. Testing will not include testing of the internals of the component; however, it will include testing to make sure that the component is invoked when it should be. Any observed responses from the component that do not match the CRD's description of its interfaces will be brought to the attention of the CCEVS validator.
7. The vulnerability analysis ETR will enumerate the types of vulnerabilities in the evaluated component that the evaluators believe could cause problems in the dependent component if they existed.
8. For the process- and guidance-based assurance classes (ACM, ADO, AGD, ALC), the ETR sections and corresponding evaluation evidence from the component evaluation will be shown to be consistent with those of the subsequent evaluation. Specifically:
    - any configuration management procedures should be applied to the component and its supporting guidance documents; the component must be uniquely and unambiguously identifiable.
    - any delivery procedures specified in the component evaluation must be shown to have been followed in getting the component to the development site of the subsequent TOE. Because this policy presumes the same developer for both, this inter-developer transfer might simply be covered under the configuration management or life-cycle procedures.
    - any guidance necessary in installing the component into the subsequent TOE (i.e. the guidance documentation for the component) must be followed.
    - while many of the requirements like development environment security may simply apply to each part of the subsequent product independently, any requirements on tools and techniques may bear additional effort since it may be the case that the tools and tool configurations have to match for the various components.
9. The ETR will remain on file with the CCEVS.
10. The VR will note that the evaluation included the results of the previous subsystem evaluation.
11. The evaluation assurance requirements for the TOE do not exceed those of the evaluated component.
12. A security analysis is conducted to demonstrate that the TOE does not adversely impact the security of the evaluated component.